VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

# Monitorování sítě s vyhledáváním anomálií

# Network monitoring with anomaly detection

2017

Jakub Večerík

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

# Zadání bakalářské práce

| | |
|---|---|
| Student: | **Jakub Večerík** |
| Studijní program: | B2647 Informační a komunikační technologie |
| Studijní obor: | 2612R059 Mobilní technologie |
| Téma: | Monitorování sítě s vyhledáváním anomálií<br>Network Monitoring with Anomaly Detection |
| Jazyk vypracování: | čeština |

Zásady pro vypracování:

Cílem bakalářské práce je monitorování sítí s vyhledáváním anomálií.

Řešení práce musí obsahovat následující body:
1. Studium a popis síťových anomálií.
2. Výběr a konfigurace linuxových programů.
3. Způsoby monitorování a vyhledávání anomálií.
4. Zátěžové testování v laboratorních podmínkách.

Seznam doporučené odborné literatury:

Bhange, A., Marhas, M. K. *Anomaly Detection in Network Traffic; A Statistical Approach: Flood and Flash Crowd Anomaly in Network Traffic* LAP LAMBERT Academic Publishing 2012, ISBN-13: 978-3659297632

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Pavel Nevlud**

Datum zadání: 01.09.2016

Datum odevzdání: 28.04.2017

doc. Ing. Miroslav Vozňák, Ph.D.
*vedoucí katedry*

prof. RNDr. Václav Snášel, CSc.
*děkan fakulty*

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne: *28. dubna 2017*

.....
podpis studenta

Rád bych na tomto místě poděkoval panu Ing. Pavlu Nevludovi za odbornou asistenci, vstřícný přístup a za konzultace, které vedly k dokončení této práce.

**Abstrakt**

Cílem bakalářské práce je zjištění funkčnosti monitorování a detekce anomálií v IPv6 síti, popsání jednotlivých typů síťových anomálií, způsoby monitorování a vyhledávání anomálií, výběr a konfigurace linuxových programů, které budou použity k vytváření a zachycování anomálií a zátěžové testování v laboratorních podmínkách, kde budou provedeny penetrační testy.

**Klíčová slova**: anomálie, detekce, monitoring, ids, ipv6, linux, síť, testování

**Abstract**

The aim of this bachelor's work is research of functioning of IPv6 Intrusion Detection System, studying and description of network anomalies, different types of ways of monitoring and searching for network anomalies, selection and configuration of linux programs which will be used for creating and detecting anomalies and stress testing in laboratory conditions, where penetration tests will be executed.

**Key Words**: anomaly, detection, ids, intrusion, ipv6, linux, network, testing

# Obsah

# Seznam použitých zkratek a symbolů

| | | |
|---|---|---|
| ACL | – | Access Control List |
| DDoS | – | Distributed Denial of Service |
| DHCPv6 | – | Dynamic Host Configuration Protocol version 6 |
| DNS | – | Domain Name System |
| DoS | – | Denial of Service |
| FTP | – | File Transfer Protocol |
| HTTP | – | Hypertext Transfer Protocol |
| HTTPS | – | Hypertext Transfer Protocol Secure |
| ICMP | – | Internet Control Message Protocol |
| IDS | – | Intrusion Detection System |
| IP | – | Internet Protocol |
| IPS | – | Intrusion Prevention System |
| IPv4 | – | Internet Protocol version 4 |
| IPv6 | – | Internet Protocol version 6 |
| ISO/OSI | – | International Standards Organization / Open System Interconnection |
| NAT | – | Network Address Translation |
| QoS | – | Quality of Service |
| R2L | – | Remote to Local |
| RA | – | Router Advertisment |
| RS | – | Router Solicitation |
| SLAAC | – | Stateless Address Autoconfiguration |
| SNMP | – | Simple Network Management Protocol |
| SSH | – | Secure Shell |
| TCP | – | Transmission Control Protocol |
| U2R | – | User to root |
| UDP | – | User Datagram Protocol |

# Seznam obrázků

# Úvod

V této části bych nejprve rozebral zadání mé práce. V následujících kapitolách se lze dočíst, co jsou to síťové anomálie, jaké jsou typy, jakými prostředky se dají vytvářet a jak je lze odhalit. Dále popíšu programy, které budou použity v této práci, od programů využitých k zachycování anomálií, přes programy monitorující síťový provoz, až po programy, které dokáží anomálie v síti nějakým způsobem vytvořit. Bude zmíněna jak funkčnost, tak základní konfigurace použitých programů. V poslední části provedu zátěžové testování v IPv6 síti, kde budou anomálie vyhledávány open source programem Snort, který monitoruje síťový provoz. Vytvořím a zachytím několik útoků, vše bude řádně okomentováno a vysvětleno, na co se vlastně u daného útoku zaměřit a co útok způsobí, respektive jestli byl úspěšně zachycen.

## 0.1 Motivace

Bezpečnost v sítích je kriticky důležitá. Pro IPv4 sítě je již mnoho návodů a videí, jak tato nastavení a testování provádět. Pro IPv6 sítě však toto téma ještě není dostatečně rozvinuto, a to bylo hlavním podnětem k dokončení této práce, tedy zjistit, jak generovat útoky v této síti a jak se proti nim bránit, protože dříve či později pozvolna dojde na plné nasazení IPv6 protokolu do všech sítí a základní nastavení systému s firewallem a antivirovým programem rozhodně není dostatečná ochrana.

# 1 Síťové anomálie a útoky na síť

Aby bylo možné provést praktické testování, je třeba nejprve zjistit jak fungují moderní sítě, jaké jsou zranitelné body v sítích a poté se seznámit s jednotlivými anomáliemi.

## 1.1 Moderní sítě

Síť je tvořena jednotlivými fyzickými a softwarovými komponenty, které poskytují služby, díky kterým lze využívat komunikačních služeb. Na síť by se dalo nahlížet jako na strukturovanou architekturu, kde hlavní role zastávají referenční model ISO/OSI a TCP/IP. Každá vrstva v modelu je zodpovědná za jiný komunikační aspekt a to tak, že neovlivňuje ostatní vrstvy. Jednotlivé vrstvy existují z důvodu strukturovaného řešení pro složitější procesy, kde každá vrstva vykonává určitou část, umožňuje komunikaci lépe pochopit a zároveň se nenarušují mezi sebou. Nicméně se zvýšenou komplexností je zde mnoho prostoru pro závady a zneužití.

## 1.2 Zranitelnost sítě

Před napadením sítě je třeba znát zranitelné body sítě. Zranitelnost sítě vychází většinou z chyb v návrhu, špatné konfigurace nebo implementace počítačových systémů a sítí. Tyto systémy mohou být nedbale nakonfigurovány a proto jsou otevřenější útokům jak zvenčí, tak zevnitř. Zranitelnost je přímo úměrná nedostatečným znalostem personálu, nedostatečné správě sítě, nedostatečnému ověřování provozu v síti a nedostačujícím bezpečnostním politikám.

### 1.2.1 Zranitelnost síťové konfigurace

- Slabá síťová architektura - velice často se stává, že společnost začíná s menší sítí a postupně se rozrůstá, nicméně bez zavedení novější architektury, která je na tento provoz lépe připravena a postupem času se začnou projevovat zastaralá řešení v sítích, kterých lze zneužít
- Nedostatečná kontrola průtoku dat - absence nebo nesprávné použití mechanismů kontroly toku dat, kdy například ACL povolí pakety z pochybných zdrojů
- Špatná konfigurace bezpečnostního vybavení a aktivních prvků - například zanechání továrního nastavení směrovače může vést k otevření portů, špatně nakonfigurovaný firewall, kdy jsou otevřeny porty, které by otevřeny být neměly
- Absence zálohy konfigurace zařízení - pokud nejsou v síti nastaveny procedury pro obnovení konfigurace síťových zařízení při nechtěném či záměrném narušení bezpečnosti sítě, ve velkých organizacích bude velmi složité nakonfigurovat vše do původního stavu a bez ztráty dat

## 1.3 Síťové anomálie

**Definice 1** *Síťová anomálie je nějaká nepravidelnost v síti, odchylka od běžného chování, která by se v síti vyskytovat neměla, většinou jde o útok za cílem odcizení dat, či narušení chodu systému*

Útoky, způsobující síťové anomálie ovlivňují zejména 2 věci:

- Výkon
- Bezpečnost

Útoky cílené na zhoršení výkonu lze pocítit například velice pomalým stahováním dat nebo se systém začne zpomalovat, dokonce může zamrznout úplně. Útoky cílené na bezpečnost mají za cíl například získat přístup k stanici jako administrátor nebo ukrást hesla více uživatelů. Anomálie se dělí na:

- infekční
- rozrůstající se
- zkoumající
- podvodné
- penetrační

První kategorie má za úkol narušit systém instalací souborů, které obsahují kód, který nějakým způsobem zapříčiní narušení systému. Typickými příklady jsou viry a červi. Rozrůstající se anomálie mají za cíl postupem času zaplnit systém různými bugy. Zahlcení bufferu patří k nejpoužívanějším.

**Definice 2** *Bug je chyba v kódu, většinou zapříčiněná programátorem.*

**Definice 3** *Buffer je vyrovnávací paměť, obsahuje dočasná data.*

Ve zkoumající kategorii se využívá sbíraní informací kvůli identifikaci slabin v síti. Například mapování portů. K podvodným útokům patří falešní uživatelé. Typickým příkladem je IP spoofing.

**Definice 4** *Spoofing označuje vznik falešné zdrojové IP adresy, což má za následek utajení totožnosti vlastníka pravé adresy.*

Penetrační útoky se spoléhají na hrubou sílu, snaží se zjišťovat všechny možné kombinace, dokud neodhalí tu správnou nebo mají za cíl zahltit systém masovými požadavky či dotazy, což má za následek zahlcení systému a ten nestíhá dotazy či požadavky zpracovávat. Typickým příkladem je DDoS.

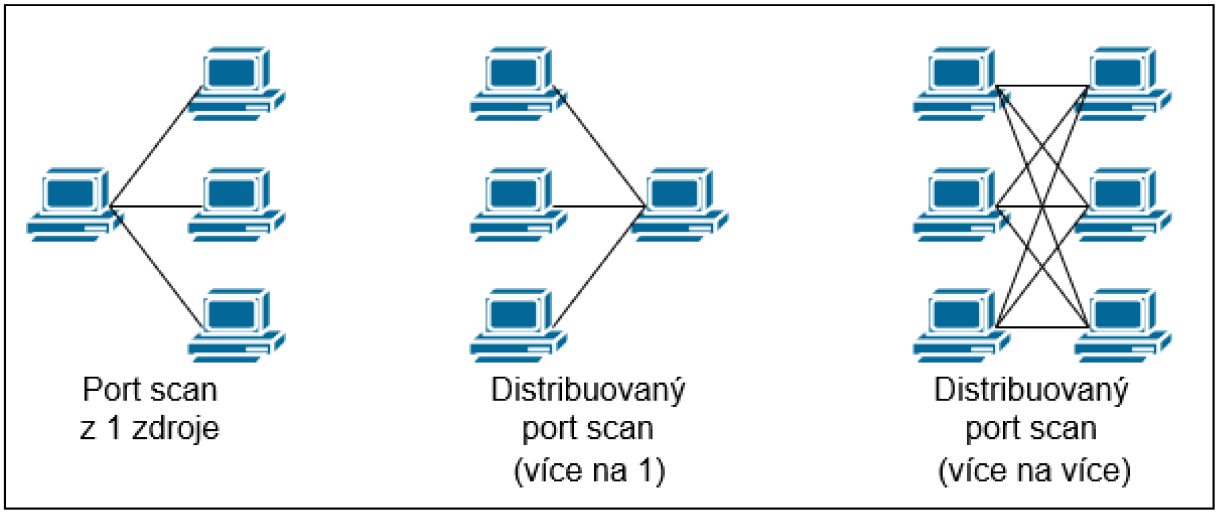
**Definice 5** *DDoS útok je takový, kdy útočník využívá více než jednu unikátní IP adresu a má za cíl způsobit nedostupnost síťových zdrojů pro uživatele v dané síti.*

## 1.4 Prostředky pro zneužívání sítě

### 1.4.1 Mapování a skenování sítě

Prostředky pro skenování sítě mají za úkol identifikovat aktivní stanice v síti za účelem útoku, či k přístupu k datům. Tyto programy podávají všeobecné informace o připojených počítačích, portech a IP adresách. Existují 3 základní typy, obrázek 1.

Obrázek 1: Typy skenování portů




Typickým zástupcem těchto programů, je Nmap. Tento program dokáže velice rychle skenovat velké sítě, dokáže identifikovat velké množství užitečných parametrů, jako například dostupné stanice, služby nabízené stanicemi, OS běžící na stanicích, či použití paketových filtrů a firewallů.

### 1.4.2 Útoky na síť

Spousta těchto programů je volně dostupná na webu. Umožňují posílat trojské koně, mapovat sítě, zkoumající útoky, DoS/DDoS a útoky na aplikační vrstvu. Velká skupina těchto programů dokáže útok specifikovat na vrstvu a protokol, takže zacílí například HTTP či FTP. Zástupcem této skupiny je program Targa, který obsahuje 16 různých programů generujících DoS útoky. Další program je například Jolt. Tento program posílá obrovské množství fragmentovaných ICMP paketů a cílový počítač je nedokáže sestavit k použití, výsledkem je zamrznutí systému a ten dále nepřijímá vstup z myši a klávesnice. A například program Panther je UDP orientovaný DoS program, který dokáže zahltit specifickou IP adresu a specifický port téměř ihned.

### 1.4.3 Falšování paketů

Tyto programy nějakým způsobem manipulují s pakety. Například program Packeth umožňuje posílat pakety se špatnou délkou hlavičky, takže pokud je správně nastaven IDS systém, paket by měl být vyhodnocen jako anomálie. Dalším zástupcem je například Packit, dovoluje generovat, monitorovat a manipulovat IP provoz. Používá se na testování NIDS, firewallu, skenování sítě a simulování provozu v síti.

### 1.4.4 Útoky na aplikační vrstvu

Při tomto útoku útočník použije například legitimní HTTP požadavek z aplikační vrstvy z legitimně připojených zařízení, aby zahltil web server. Tento útok je mnohem rafinovanější, než klasický DoS útok, protože útočník používá legitimní protokoly a spojení. Odhalit tento typ útoku je velká výzva. Například útoky na SMTP protokol zahlcují e-mail a přenášejí SMTP červy.

### 1.4.5 Fingerprinting útoky

Jsou určeny k identifikaci specifických dat síťového protokolu analýzou jeho vstupu a výstupu. Za tyto data se považují verze protokolu či konfigurovatelné parametry. Tyto útoky se používají hlavně kvůli identifikaci operačního systému běžícího na vzdálené ploše. Síťoví správci mohou používat vzdálený fingerprinting ke sběru informací a k usnadněnému spravování sítě. Programy podporující tento způsob útok jsou například Nmap či Queso. Druhý zmíněný dokáže vzdáleně určit operační systém, jeho verzi a výrobce analýzou paketů. Tento program udává přesné informace o síti či systému skenováním sítě.

### 1.4.6 Uživatelské útoky

Při těchto útocích se útočník vydává za běžného uživatele a snaží se získat administrátorská práva či přístup k lokálnímu počítači bez založeného účtu. Oba pokusy jsou velice náročné na zjištění, neboť tento útok reflektuje běžné chování uživatele. Jsou 2 způsoby: U2R útok a R2L útok. V prvním zmíněném se útočník snaží získat přístup k lokálnímu počítači, například vystopováním hesla. Poté se útočník pokusí využít slabin operačního systému a získat administrátorská práva. Jakmile útočník získal práva, nainstaluje backdoor nebo nějakým způsobem zmanipuluje soubory operačního systému. Program pro tyto účely je například Yaga. Tento program vytvoří nový administrátorský účet, zneužitím registrových souborů. Útočník upraví registry aby shodil některé systémové služby a vytvořil si administrátorský účet. Při R2L útocích se útočník pokouší získat přístup bez účtu na cílovém počítači. K takovému útoku útočník používá například e-mail, ve kterém se nachází backdoor pomocí kterého se poté útočník do počítače dostane. Program pro tyto účely je například Netcat. Tento program nainstaluje trojského koně a spustí Netcat na portu 53, tedy DNS. Útočník poté využije Netcat port pro přístup do počítače bez uživatelského jména a hesla.

# 2 IPv6 vs IPv4

## 2.1 Zápis adres

IPv6 protokol byl primárně vyvinut, aby vyřešil problém s nedostatkem adres protokolu IPv4. Jen krátce rozdíly v zápisu:

- IPv4 adresa - 192.168.0.1/24
- IPv6 adresa - 2001:db8:ab:2::1/64

Zápis je tedy naprosto jiný. IPv4 adresa se skládá ze 4 rozdělených oktetů a za lomítkem počet bitů vyhrazených pro podsíť. IPv6 adresa se skládá z 8 skupin po 16 bitech, zápis masky neexistuje, místo toho se používá tzv. prefix, který se zadává za lomítkem IP adresy. IPv4 adresa má tedy velikost 32 bitů, zatímco IPv6 128 bitů. Samozřejmě, i IPv6 obsahuje vyhrazené adresy, některé z nich:

- fe80::/10 - linková lokální adresa - validní pouze v lokální síti, nelze směrovat, využívá se při DHCPv6 a NDP
- ff00::/8 - multicast adresa - pokud je odeslán paket na tuto adresu, je hned známo, že se jedná o multicast
- ::1/128 - local host adresa - obdoba 127.0.0.1

## 2.2 NAT

Další rozdíl oproti IPv4 je NAT. Kde IPv4 využívá NAT hlavně proto, protože se šetří adresami a protože si lze skrýt celou podsíť za jedinou adresu (toto řešení znemožňuje či komplikuje komunikace některých softwarů a nelze se snadno připojit k jinému zařízení za NATem), IPv6 NAT nepotřebuje, protože adres je velká spousta a lze je dynamicky velice rychle měnit, takže zjištěná adresa útočníkem už při útoku dávno nemusí být platná.

## 2.3 Konfigurace

Jedna z dalších významných změn, je konfigurace adresy. Lze toho docílit pomocí dvou metod, a to:

- bezstavová autokonfigurace (SLAAC)
- DHCPv6

První zmíněna metoda funguje tak, že nevyžaduje informace od DHCP serveru, ale zařízení si vytvoří IPv6 adresu na základě informací, které má už k dispozici, například linková adresa. Tímto způsobem je řešena hostitelská část adresy. K identifikaci sítě slouží tzv. prefix. SLAAC

funguje tak, že směrovač zařízením v pravidelných intervalech oznamuje, v jaké síti se nacházejí a které pakety mají putovat z naší sítě. To se nazývá Router Advertisement (RA). Nově připojené zařízení vysílají do sítě požadavek Router Solicitation (RS) s žádostí o informace v jaké síti se nacházejí a kudy se dostat ven. Veškerá komunikace probíhá pomocí ICMPv6. Druhá metoda, tedy DHCPv6 obsahuje 2 režimy:

- bezstavové DHCPv6
- stavové DHCPv6

Bezstavové DHCPv6 je v podstatě nadstavba SLAAC. Zařízení obdrží od směrovače příznaky M - managed a O - other. Pokud je nastaven příznak M, použije se stavové DHCPv6, pokud O, bude použito bezstavové DHCPv6, pokud jsou oba příznaky vynulované, v síti není k dispozici DHCPv6 server. Funguje to tedy tak, že klient vyšle do sítě RS, směrovač odpoví - RA, klient si nakonfiguruje parametry rozhraní a podle příznaků RA odešle DHCPv6 požadavek. Stavové DHCPv6 je více podobné DHCP používanému v IPv4 sítích. Klient požádá DHCP server o přidělení adresy pomocí DHCPv6, příznak M, adresa je na určitou dobu přidělena klientovi a přijetí adresy je potvrzeno. Nicméně klient může využívat i adresy, které získal od směrovače pomocí RA požadavku. Výchozí bránu nelze získat pomocí DHCPv6, ale pouze pomocí oznámení směrovače - RA.

## 2.4 Bezpečnost

V IPv6 byla původně povinná implementace bezpečnostních mechanismů IPsec, nicméně toto nařízení bylo zmírněno a odloženo na později, IPsec je tedy momentálně dobrovolný. Tyto opatření jsou:

- Authentication header
- Encapsulating Security Payload

První zmíněné opatření slouží k ověření totožnosti odesílatele datagramu a správnosti obsahu. Lze také zamezit opakovanému zasílání paketů. Druhé zmíněné opatření má za úkol zašifrovat datagram. Tyto prvky mohou být uplatněny ve 2 režimech, a to:

- Transportní režim
- Tunelující režim

V transportním režimu jsou bezpečnostní hlavičky vloženy odesílatelem datagramu mezi ostatní rozšiřující hlavičky. V tunelujícím režimu je celý datagram zabalen jako data nového datagramu v jehož hlavičkách se nacházejí bezpečnostní prvky.

# 3 Způsoby monitorování a vyhledávání anomálií

## 3.1 Monitorování sítě

Aby bylo možné anomálie odhalit, je třeba monitorovat síť. Způsobů jak monitorovat síť je velká spousta. Je třeba si uvědomit, co přesně je třeba v síti sledovat. Lze monitorovat bez agenta, ale mnohem efektivnější způsob je monitoring s agentem a pro menší sítě je nejlepší, když konfiguraci provádí člověk, protože to zaručí větší přehled o událostech, které se dějí v síti.

## 3.2 Oblasti pro monitoring

Obecně se oblasti pro monitoring dělí na:

- servery a jejich služby
- aktivní síťové prvky
- síťová komunikace, provoz
- bezpečnost

Bližší specifikace oblastí při monitorování provozu v síti:

- dostupnost serverů, služeb, aplikací
- události na serverech
- vytížení zdrojů
- vytížení linek
- statistiky síťového provozu
- analýzy nestandardního chování v síti
- informace o portech
- bezpečnostní incidenty

Je tedy třeba si uvědomit, co přesně je třeba sledovat při různých událostech v síti, protože každý útok je cílen na jinou oblast. Na základě těchto poznatků je třeba zvolit bezpečnostní opatření.

## 3.3 Technologie pro monitoring

Jak už bylo řečeno, lze tedy monitorovat s agentem, či bez agenta. Při monitorování bez agenta se testují vlastní služby serveru nebo se data získávají pomocí standardních protokolů, například SNMP (součást sady internetových protokolů). Ne příliš efektivní metoda pro větší sítě. Monitorování s agentem, tedy způsob, který je popisován v této práci, je založen na konfiguraci nějakého klienta. Tento způsob umožňuje získávat podrobnější údaje a nastavit komplexnější ochranu sítě. Technologie pro monitoring sítě jsou:

- dostupnost serverů pomocí ping
- dostupnost služby - navázaní TCP spojení
- události ze serverů - syslog
- získávaní dat pomocí klienta
- získávaní údajů pomocí protokolů
- sledování síťového toku
- analýza síťových protokolů
- bezpečnost v síti IDS/IPS

## 3.4 Způsoby vyhledávání anomálií

Systémy Monitorující anomálie, respektive IDS se dělí na 2 základní skupiny:

- HIDS - Host based IDS
- NIDS - Network IDS

První zmíněný způsob jen okrajově, jelikož se tato práce zaměřuje na NIDS. HIDS je tedy orientovaný na hostitelský systém. Tyto systémy využívají záznamy generované jádrem operačního systému, monitorují probíhající procesy v kontextu se spuštěnými aplikacemi a změnami v souborech. NIDS zpracovávají informace získané ze síťových rozhraní, kritické je jejich umístění, aby zachytily co největší síťový provoz. Mezi jejich výhody patří

- při dobrém rozmístění možnost monitoringu velké sítě
- NIDS nijak neovlivňují provoz sítě

Mezi jejich slabší stránky patří

- může být obtížné zpracování všech paketů při velkém provozu
- nelze analyzovat šifrovaný provoz
- nelze přehledně zjistit, zda byl útok vedený na síť kompletní

## 3.5 Metody detekce anomálií

### 3.5.1 Porovnávání signatur

Využívá se u NIDS systémů, signatury popisují nějaké specifické chování při známém útoku a na základě toho NIDS dokáže detekovat útok a změnit pravidla firewallu, či Quality of Service.

**Definice 6** *Quality of Service (QoS) je celkový výkon v telefoniích či počítačových sítích, respektive výkon, který dorazí k uživateli sítě*

Tato metoda se používá při monitoringu sítě na úrovni paketů, nicméně pokud na systém dorazí útok, který není znám vzorcem v databázi signatur, systém jej nerozpozná.

### 3.5.2 Analýza chování

Metoda, která je závislá na nějakém vzoru provozu na síti a poté porovnává reálný provoz v síti s vytvořeným vzorem provozu v pravidelných časových intervalech. Údaje vybočující z hranic systém detekuje jako anomálie.

### 3.5.3 Stavová analýza

Je založena na přesných definicích protokolů vyskytujících se v síti. Činnost protokolu je dána a přechody mezi stavy má za úkol stavový automat. Pokud tedy provoz v síti neodpovídá stavu definovaném v protokolu, tento stav je vyhodnocen jako anomálie.

### 3.5.4 Učení pod dohledem

Při této metodě se vybuduje třídní prediktivní model pro normální a nestandardní chování. Nové instance dat jsou testovány, aby se začlenily do nějaké třídy. Nicméně nastávají 2 základní problémy, a to:

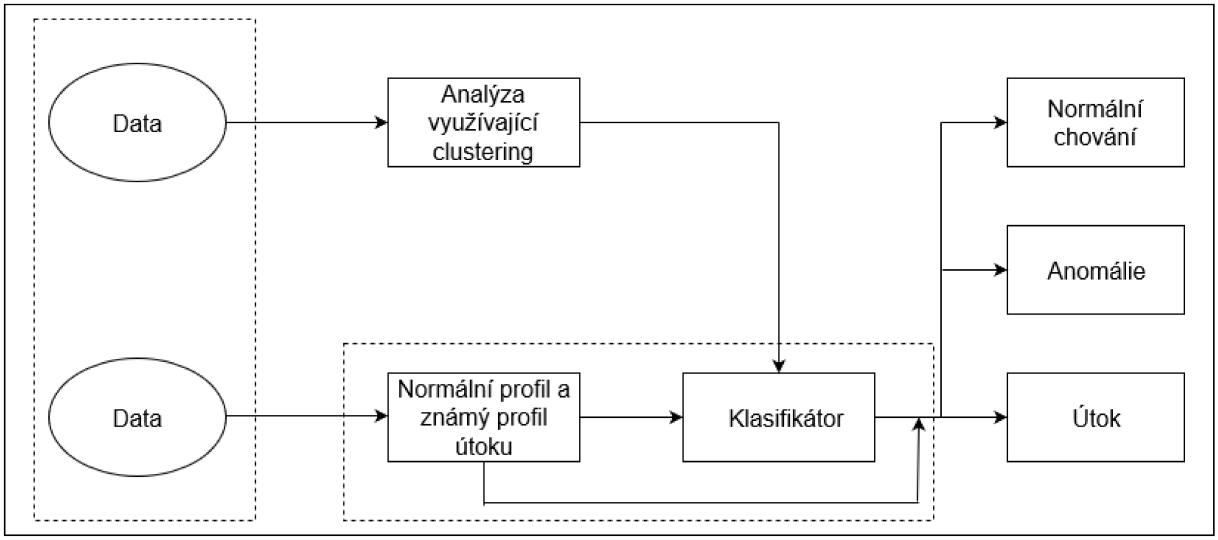
- počet instancí běžného chování je při testování většinou vždy více, než nestandardního
- získávání přesného zařazení do tříd hlavně pro nestandardní chování je velice složité

### 3.5.5 Metoda využívající Clustering

**Definice 7** *Clustering je metoda analýzy dat, při které se data zařazují do skupin, které si jsou více podobné, aby bylo možno data přehledně rozlišovat*

Při této metodě se tedy data rozdělují do skupin, obrázek 2.

Obrázek 2: Schéma sítě, na které proběhly testy




Pro každou skupinu se zvolí reprezentativní bod na základě kterého budou data přidávána právě do dané skupiny. Není třeba explicitní popis daných tříd, či typů anomálií systémovým administrátorem, protože toto řešení je realizováno tak, že se systém sám dokáže učit a rozdělovat anomálie. Nejsou třeba ani žádná zkušební data.

### 3.5.6 Detekce anomálií na základě pravděpodobnosti

Fungují tak, že ohodnocují výstup dat systémů afektovaných náhodnými jevy nebo dalšími typy pravděpodobných jevů. Hlavní charakteristikou této metody je schopnost upravit předešlou predikci výstupu na základě porovnání s novými daty. Metod v tomto odvětví je více, jedna z nich, Naive Bayes Methods funguje tak, že počítá pravděpodobnost výsledku z několika dodaných proměnných obsahujících vzájemně spojená data. Tento algoritmus počítá pravděpodobnost výsledku pro specifický atribut a poté tuto pravděpodobnost uloží. Tato operace je provedena pro všechny atributy. Čas, který byl potřeba k spočítání pravděpodobnosti pro danou třídu pro každý případ je v nejhorším případě úměrný počtu dodaných atributů, na základě kterých byl výpočet proveden.

### 3.5.7 Strojové učení

**Definice 8** *Strojové učení je oblast v počítačových vědách, kde se počítač za chodu "učí"bez toho, aniž by byl explicitně programován. Na základě získaných dat je schopen "rozhodovat se"a "předpokládat"různé situace.*

Počítač jako takový udržuje informace o tom kdo na něm provádí nějaké aktivity a odkud. Například informace o uživateli, který zaslal odněkud soubor do nějakého cíle. Existují zařízení, které monitorují každý střípek dat, který do počítače přijde, či z něj odejde. Na základě toho, je schopen odhalit nějakou anomálii, například aktivitu, která neměla být vykonána a přesto vykonána byla. Tyto aktivity mohou znamenat narušení sítě. Abnormální aktivity lze porovnávat se známými instancemi anomálií. Nicméně vzorce narušení nemusí být úplně jednoznačné, či dokonce jednoduché k nalezení. Na základě toho vzniklo strojové učení. Stroj se snaží získat validní a potenciálně užitečná data a vzorce, většinou velkých objemů.

# 4 Výběr a konfigurace linuxových programů

V této kapitole budou představeny a popsány programy, které byly vybrány pro testování v laboratoři. Jsou to programy pro monitorování provozu v síti, vyhledávání anomálií a pro vytvoření anomálií. Základní konfigurace byla provedena v domácí IPv4 síti, nicméně všechny zmíněné programy byly použity pro zátěžové testování ve školní laboratoři, tudíž je zaručena funkčnost v IPv6 síti.

## 4.1 Snort IDS

Tento program je nejrozšířenější NIDS program, který je zadarmo. Má velkou podporu komunity, existují přehledné dokumentace a video návody. V prvé řadě je ale velice efektivní při detekci anomálií jak v malých tak ve velkých sítích a obsahuje podporu IPv6.

### 4.1.1 Instalace

Před instalací je doporučeno stáhnout všechny potřebné knihovny a mít systém v nejnovější verzi, tím se lze vyhnout potenciálním chybám:

```
apt-get install -y build-essential
apt-get install -y libpcap-dev
apt-get install libpcre3-dev
apt-get install -y libdumbnet-dev
apt-get install zlib1g-dev
apt-get install bison flex
sudo apt-get update
```

Instalaci je možno učinit pomocí příkazů z oficiálního webu Snort, nebo lze stáhnout z Ubuntu repozitáře pomocí:

```
sudo apt-get install snort
```

Během instalace se program zeptá na rozhraní, na kterém má sledovat provoz a na IP adresu podsítě, nicméně tyto parametry lze konfigurovat kdykoliv po instalaci v souboru snort.conf nebo zadávat přímo do terminálu při spouštění Snortu.

### 4.1.2 Základní konfigurace

Program je tedy nainstalován, nyní je potřeba jej nakonfigurovat. Začal jsem tedy potřebnou editací konfiguračního souboru snort.conf. Je potřeba nastavit laboratorní síť a externí síť (co se externí sítě týče, v této práci nebylo potřeba specifikovat, neboť byly útoky prováděny z vnitřní sítě), v konfiguračním souboru začíná 45. řádkem:

```
ipvar HOME_NET 2001:db8:ab:2::/64
ipvar EXTERNAL_NET !$HOME_NET
```

Dále je třeba nastavit cestu k pravidlům, například přiřazením cesty do proměnných:

**Definice 9** *Pravidla jsou ve Snortu kritickou položkou, neboť díky nim program zachytí anomálii. Pravidla lze tvořit samostatně, ale velká spousta je již vytvořena komunitou. Fungují tak, že pokud paket splňuje podmínku/y pravidla, bude provedena činnost definovaná daným pravidlem, tedy například, pokud se někdo neoprávněně připojí přes SSH k mému počítači a pravidlo je správně napsáno, Snort paket zachytí a provede mnou zadefinovanou činnost.*

```
var RULE_PATH /etc/snort/rules
var SO_RULE_PATH /etc/snort/so_rules
var PREPROC_RULE_PATH /etc/snort/preproc_rules
var WHITE_LIST_PATH /etc/snort/rules
var BLACK_LIST_PATH /etc/snort/rules
```

Základní konfigurace je tedy hotova. Velká spousta parametrů se dá nastavit v terminálu při spouštění Snortu.

### 4.1.3 Režimy

**4.1.3.1 Sniffer režim** V tomto režimu Snort do terminálu vypisuje IP adresy a TCP, UDP či ICMP hlavičky. Stačí zadat:

```
sudo snort -v
```

Pro detailnější výpis, včetně dat paketů slouží:

```
sudo snort -vd
```

Pro ještě specifičtější výpis, to je i s hlavičkou linkové vrstvy, je třeba použít:

```
sudo snort -vde
```

**4.1.3.2 Packet logger režim** Umožňuje všechna data zaznamenávat na disk, je třeba specifikovat log adresář, poté stačí zadat:

```
sudo snort -dev -l /var/log/snort
```

Jakmile jsou pakety zapsány do binárního souboru, lze pakety přečíst jakýmkoliv softwarem, který podporuje tcpdump binární formát, ale zvládne to i Snort přímo do terminálu:

```
sudo snort -r jmeno_souboru
```

Snort dokáže filtrovat specifické pakety, například jen UDP:

```
sudo snort -dvr packet.log udp
```

**4.1.3.3 NIDS režim** NIDS režim je určitě nejdetailnější a nejpoužívanější režim, monitoruje provoz v síti v reálném čase a podle definovaných pravidel provede akci. Pro zapnutí v NIDS režimu je třeba specifikovat cestu, ve kterém se nachází hlavní konfigurační soubor snort.conf. Konfigurační soubor je již připraven. Soubory, na kterých teď záleží, jestli bude anomálie odhalena nebo ne, mají koncovku rules. Takzvaná pravidla se píší tímto způsobem:

```
Akce Protokol ZdrojIP ZdrojPort -> CilIP CilPort (volby)
```

Jako akci lze zadat tyto možnosti:

- alert - zapíše varování do souboru alert
- log - vypisuje specifikované pakety do souboru log
- pass - propustí paket
- drop - zahodí paket

Do protokolu lze zadat:

- TCP
- UDP
- ICMP
- IP

Do cílové a zdrojové IP adresy lze zadat jak IPv4 adresu, tak IPv6 bez jakékoliv další konfigurace. Snort plně podporuje IPv6 protokol. Jako cílový a zdrojový port lze zadat jakýkoliv, podle toho, co má Snort vyhledávat, například HTTP port 80, či DNS port 53. Poslední částí v pravidlu jsou volby, do nich lze psát metadata, některá z nich nemají žádný vliv na detekci a slouží pro lepší identifikaci či vyhledání, lze zadat například:

- msg - vypíše zadefinovanou zprávu
- sid - tzv. Snort ID, jednoznačný identifikátor pravidla, je povinný
- rev - obsahuje číslo revize pravidla
- priority - udává prioritu paketu, podle toho, jak vysoká priorita byla anomálii přiřazena

- count a seconds - count značí, kolikrát musí daná situace nastat a seconds za jaký čas, například count 5, seconds 10 (pokud nastane situace zadefinovaná v pravidlu 5x za 10 sekund, je vyhodnocena jako anomálie)

Dále lze specifikovat data obsažená v paketu:

- content - řetězec, který bude v paketu hledán, rozlišuje malá a velká písmena
- nocase - hledání řetězce, nerozlišuje malá a velká písmena
- depth - určuje jak daleko bude hledat v datové části paketu

a v neposlední řadě data obsažená v IP hlavičce:

- ttl - hodnota Time to live
- flags - flags příznaky v TCP hlavičce, např S-SYN, A-ACK
- ip proto - číslo IP protokolu

Podrobný popis všech možností při psaní pravidel je uveden na oficiálních stránkách Snortu, které jsou uvedeny ve zdrojích.

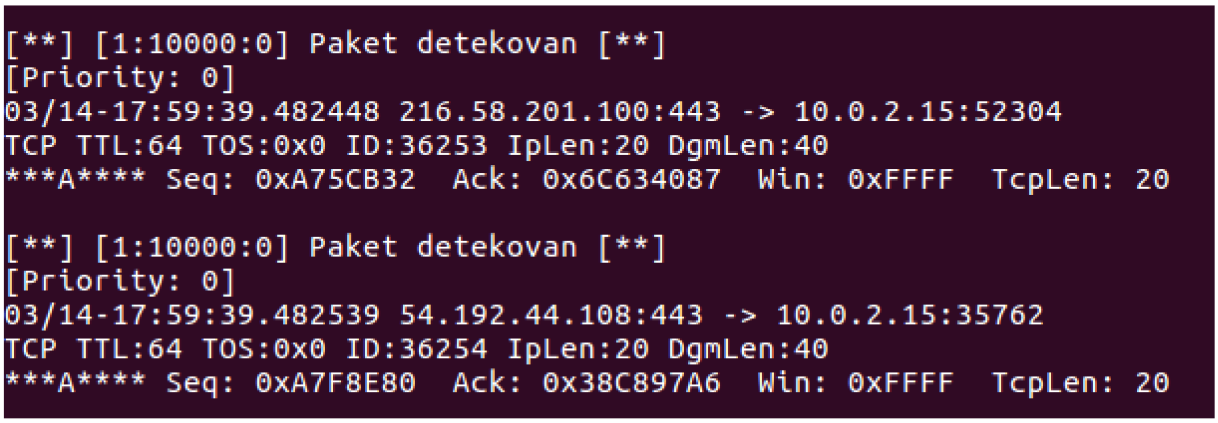
### 4.1.4 Psaní pravidel

Jak bylo řečeno, vše záleží hlavně na pravidlech. Paket je zachycen a vyhodnocen jako anomálie nebo běžný provoz podle zadefinovaných pravidel. Zde příklad jednoduchého pravidla, které zachytí veškerý provoz v síti:

```
alert IP any any -> any any (msg:"Paket detekovan"; sid:10000;)
```

Obrázek výstupu alert souboru 3.

Obrázek 3: Test zachytávání veškerého provozu do souboru alert



Například pro detekci neautorizovaného přihlášení z externí sítě přes SSH lze napsat:

```
alert IP !IP_hlidane_site any -> IP_hlidane_site 22 (msg:"Neopravnene pripojeni
    pres SSH"; sid:10001;)
```

Některé programy byly zkoušeny na IPv4 síti, nicméně všechny zmíněné programy byly použity v laboratorním testování na IPv6 síti, takže podpora IPv6 byla ověřena.

## 4.2 Nmap

Program sloužící k mapování otevřených portů a zjišťování operačního systému. je také zdarma. Instalaci lze opět provést stažením z Ubuntu repozitáře, konfigurace se neprovádí, všechny argumenty se zadávají do příkazové řádky. Příklad mapování s parametry o zjištění operačního systému, obrázek 4:

```
nmap -A -Pn 192.168.0.14
```

Obrázek 4: Výsledek skenování portů a zjišťování operačního systému

```
Not shown: 996 closed ports
PORT     STATE SERVICE      VERSION
135/tcp  open  msrpc        Microsoft Windows RPC
139/tcp  open  netbios-ssn  Microsoft Windows 98 netbios-ssn
445/tcp  open  microsoft-ds Microsoft Windows 10 microsoft-ds
5357/tcp open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
eOffice Writer
cpe:/o:microsoft:windows_98, cpe:/o:microsoft:windows_10

Host script results:
| smb-os-discovery:
|   OS: Windows 10 Pro 14393 (Windows 10 Pro 6.3)
|   OS CPE: cpe:/o:microsoft:windows_10::-
|   NetBIOS computer name: DESKTOP-02RQ4PE
|   Workgroup: WORKGROUP
```

Je tedy vidět, že nmap úspěšně odhalil operační systém na mém počítači.

## 4.3 Slurm

Program, monitorující provoz v síti na zadefinovaném síťovém rozhraní. Slurm je zdarma a lze ho stáhnout z Ubuntu repozitáře. Program byl využit pro přehledné zobrazení příchozích a odchozích paketů. Pro zobrazení provozu na rozhraní stačí zadat:

```
sudo slurm -i eth0
```

Zde výstup například při prohlížení internetu 10.

## 4.4 Speedometer

Další použitý program pro přehledné zobrazení rychlosti v grafové podobě, použit také za účelem porovnání provozu při běžných činnostech a při útocích. Pro monitorování rychlosti lze zadat:

```
sudo speedometer -i eth0
```

Zde vizualizace rychlosti přenosu při stahování dat z internetu 16.

## 4.5 Linuxová distribuce Backtrack 5 R3

Tato distribuce Linuxu, přístupná zdarma, byla použita za účelem generování reálných útoků. Byla využita pro DoS a útok hrubou silou. Obsahuje programy, které plně podporují útoky na IPv6 síť, nicméně jich není ani zdaleka tolik, jako pro IPv4 sítě. Pro DoS útok byl využit router flooding a pro útok hrubou silou program Hydra.

## 4.6 Packet sender a Packeth

Tyto programy byly využity k vytvoření potřebného provozu pro zachycení anomálie Snortem. Program Packeth se však v IPv6 síti příliš neosvědčil, vždy po vygenerování (někdy i předtím) program spadl. Program Packet sender byl využit pro vygenerování UDP paketů 22, avšak TCP nebo ICMP pakety nebylo možné úspěšně odeslat. Funkcionalita v IPv6 sítích je tedy stále značně omezena.

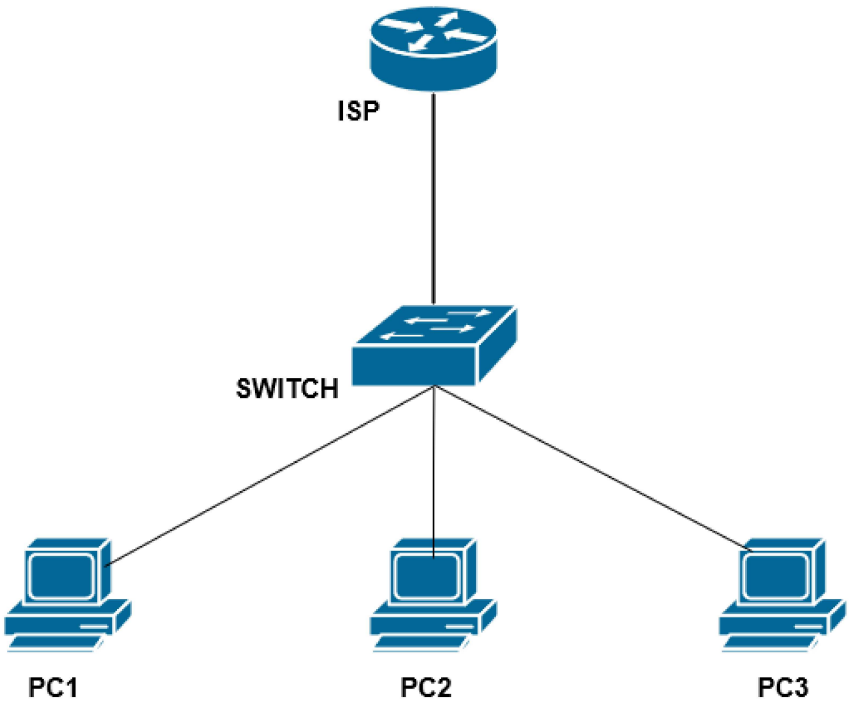
# 5 Zátěžové testování v laboratorních podmínkách

V této kapitole je popsáno zátěžové měření, které proběhlo ve školní laboratoři EB215, kde je funkční síť pouze s podporou IPv6. Testování proběhlo na 3 počítačích připojených ve stejné síti. Toto zapojení bylo vybráno pro demonstraci toho, že útoky nemusí v žádném případě přicházet zvenčí. Schéma zapojení sítě na obrázku 5. Počítače byly nakonfigurovány pomocí následujících příkazů (příklad pro PC1):

```
sudo stop network-manager
sudo ip addr add 2001:db8:ab:2::1 dev eth0
sudo ip link set dev eth0 up
```

Network manager byl zastaven, aby bylo možno bezproblémově přidat statickou IP adresu, druhý příkaz je pro přidání IP adresy na rozhraní eth0, třetí příkaz rozhraní eth0 nahodí a jsou přiřazeny IP adresy pro komunikaci s internetem. IDS Snort byl nainstalován na PC s IP adresou 2001:db8:ab:2::1, ale jako monitorovanou oblast jsem zadal celou síť, tedy 2001:db8:ab:2::/64, protože je třeba chránit celou síť a to i před případnými útoky, či pokusy o zneužití bezpečnosti zevnitř.

Obrázek 5: Schéma sítě, na které proběhly testy, PC1 - 2001:db8:ab:2::1, PC2 - 2001:db8:ab:2::2, PC3 - 2001:db8:ab:2::3



## 5.1 Neoprávněné připojení přes SSH

Toto neoprávněné připojení bylo zvoleno pro demonstraci zanedbání bezpečnosti v síti. Politika v síti je nastavena tak, aby se počítače v síti vzájemně nemohly připojovat na sebe. Přece jen SSH slouží hlavně k připojení na vzdálený počítač, většinou za účelem využití služeb, kterými můj počítač nedisponuje. Pravidlo ve Snortu bylo napsáno tak, aby zachytilo jakékoliv připojení přes SSH na kterýkoliv počítač v testované síti.

```
alert tcp $HOME_NET any -> $HOME_NET 22 (msg:"Neopravnene pripojeni pres SSH v
    nasi siti"; sid:10030;)
```

Snort byl spuštěn následujícím příkazem (tento příkaz byl použit u všech testů, proto jej uvedu jen zde):

```
sudo snort -dev -h 2001:db8:ab:2::/64 -l var/log/snort -c /etc/snort/snort.conf
     -A full -i eth0
```

Připojení bylo provedeno z počítače s IP adresou 2001:db8:ab:2::2 na počítač s IP adresou 2001:db8:ab:2::1.

```
ssh -6 student@2001:db8:ab:2::1
```

Na obrázku 6 lze vidět Snortem zachycený paket.

Obrázek 6: Zachycený paket Snortem při připojení přes SSH

```
[**] [1:10030:0] Neopravnene pripojeni pres SSH v nasi siti. [**]
[Priority: 0]
03/28-13:35:53.943059 74:D4:35:7C:29:F0 -> 74:D4:35:73:2B:73 type:0x86DD len:0x56
2001:db8:ab:2::2:42160 -> 2001:db8:ab:2::1:22 TCP TTL:64 TOS:0x10 ID:0 IpLen:40 DgmLen:72
***A**** Seq: 0x72FE73C2  Ack: 0xDB36FF8A  Win: 0x4BB  TcpLen: 32
TCP Options (3) => NOP NOP TS: 3333308 3334302
```

IP adresy odpovídají, tedy 2001:db8:ab:2::2 jako zdrojová adresa, 2001:db8:ab:2::1 jako cílová adresa, připojení na port 22, tedy SSH.

## 5.2 Port scan útok

Aby útočník zjistil, které porty jsou na počítači otevřené, musí provést skenování portů. Podle zadaných argumentů se bude lišit výstup skenování. V mém případě bylo cílem zjistit otevřené porty, pomocí FIN skenování.

**Definice 10** *FIN skenování je typ skenování portů, který dokáže obejít některé bezstavové firewally a routery. Tyto firewally zakazují příchozí TCP provoz, zatímco povolují odchozí.*

Skenování portů proběhlo z počítače s IP adresou 2001:db8:ab:2::2 na počítač 2001:db8:ab:2::3.

```
sudo nmap -6 -sF 2001:db8:ab:2::3
```

Pravidlo ve Snortu bylo zadefinováno následovně:

```
alert tcp any any -> $HOME_NET any (msg:"PC v mistni sisi je skenovan"; flow:
    stateless; gid:100020 sid:100020; rev:4;)
```

Zápis stateless se používá právě při skenování portů, či útoků, které mají za cíl znemožnit správnou funkci počítače. V podstatě to znamená, že tyto pakety neobsahují žádnou informaci o navázání TCP spojení. Tento útok byl opět úspěšně zachycen Snortem, čímž byla potvrzena funkčnost skenování portů v IPv6 síti. Výstup zachycený Snortem na obrázku 7.

Obrázek 7: Zachycený paket Snortem při mapování portů

```
[**] [1000020:1000020:4] PC v mistni siti je skenovan [**]
[Classification: Attempted Information Leak] [Priority: 2]
03/28-13:26:06.257843 74:D4:35:73:2B:73 -> 74:D4:35:7C:2D:7D type:0x86DD len:0x4A
2001:718:1001:2c8:bcc1:39fb:e486:5f71:34689 -> 2001:db8:ab:2::3:199 TCP TTL:255 TOS:0x0
*******F Seq: 0x28A735C7  Ack: 0x0  Win: 0x400  TcpLen: 20
```

Zdrojová IP adresa je jedna z přiřazených na rozhraní eth0 u počítače 2001:db8:ab:2::2, cílová už odpovídá PC3, tedy 2001:db8:ab:2::3. Skenováním byl zjištěn jediný otevřený port, a to SSH - 22, obrázek 8. Porty lze otevírat a zavírat ručně v nastavení firewallu.

Obrázek 8: Skenování portů PC s IP adresou 2001:db8:ab:2::3

```
Nmap scan report for 2001:db8:ab:2::3
Host is up (0.00014s latency).
Not shown: 999 closed ports
PORT   STATE         SERVICE
22/tcp open|filtered ssh
MAC Address: 74:D4:35:7C:2D:7D (Unknown)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 2.79 seconds
           Raw packets sent: 1088 (65.292KB) | Rcvd: 1085 (65.112KB)
```

## 5.3 Monitoring běžného provozu v síti

Abych byl schopen rozlišit nestandardní chování v síti, nejprve jsem provedl testy přenosové rychlosti a vizualizaci příchozích a odchozích paketů pomocí programů Slurm a Speedometer.

### 5.3.1 Surfování na internetu

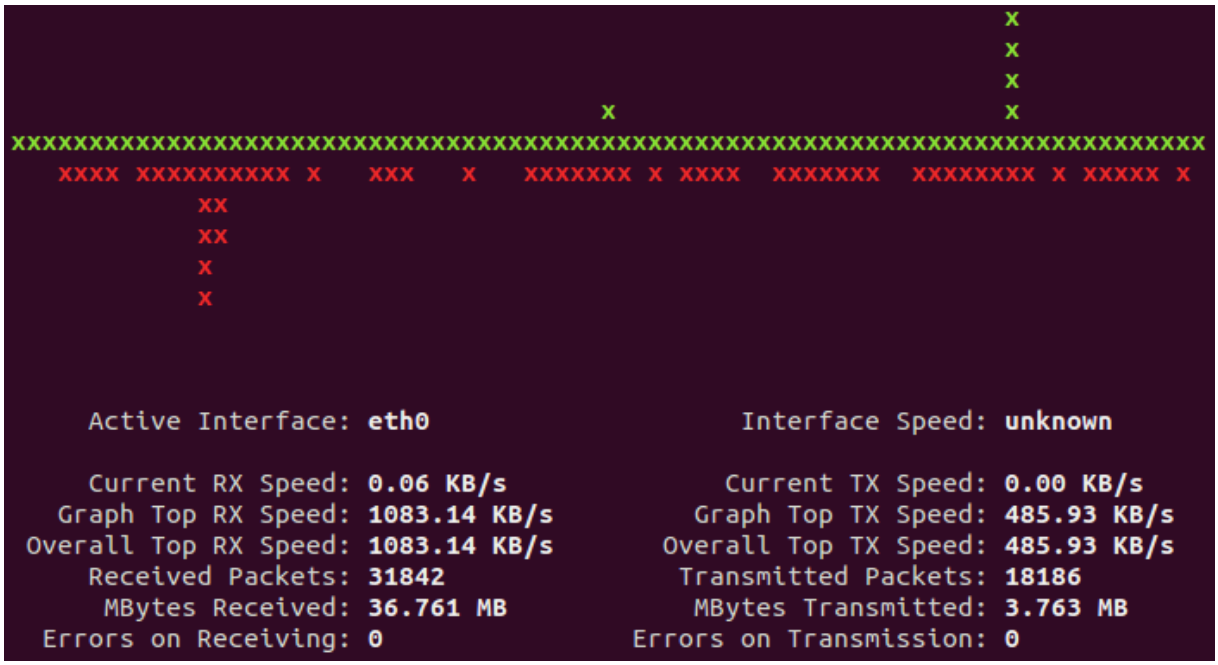
Běžný síťový provoz při surfování na internetu tedy vypadá z pohledu Snortu následovně 9.

Obrázek 9: Pakety zachycené Snortem při surfování na internetu, IP adresy pochopitelně neodpovídají adresám vnitřní sítě, ale veřejným adresám přiděleným školním DHCPv6

```
04/05-13:06:54.503725 2001:718:1001:2c8:e4cc:566a:5b6b:7573:42337 -> 2001:718:10
01::53:53
UDP TTL:64 TOS:0x0 ID:0 IpLen:40 DgmLen:89
Len: 41
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/05-13:06:54.504180 2001:718:1001::53:53 -> 2001:718:1001:2c8:e4cc:566a:5b6b:7
573:42337
UDP TTL:62 TOS:0x0 ID:0 IpLen:40 DgmLen:293
Len: 245
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

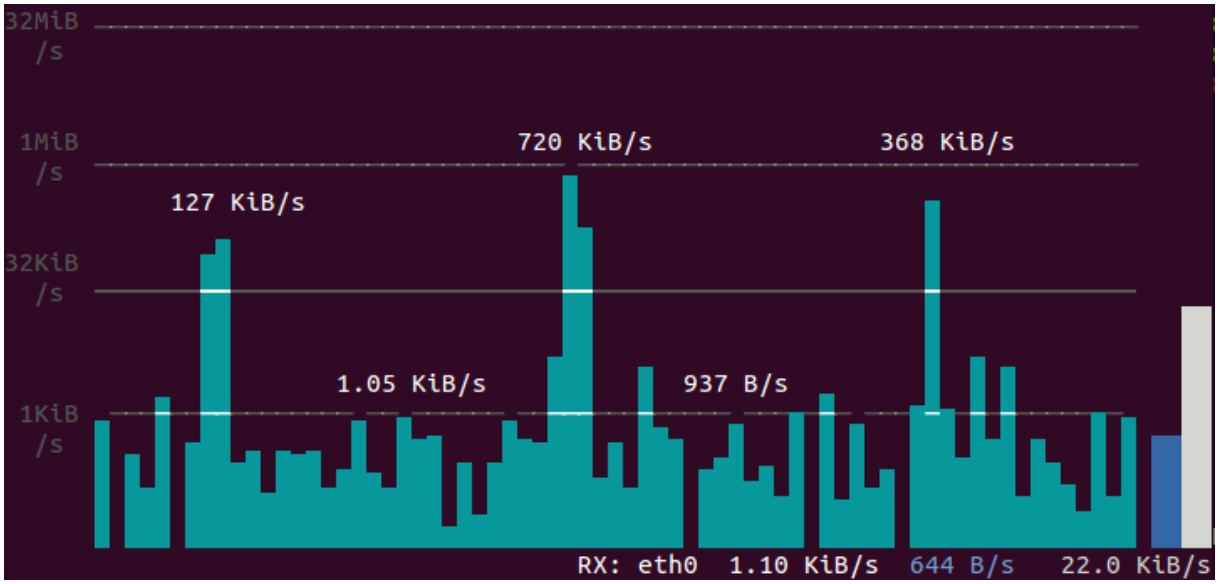
Na tomto provozu není nic neobvyklého, adresy se dají dohledat pomocí IP lookup programů, obě adresy jsou VŠB adresy, jsou validní, pakety nepřišly v neobvyklém množství ani v neobvyklých intervalech. Pokud se na pakety podívám z pohledu odchozích a příchozích, také neuvidím nic neobvyklého, obrázek 10.

Obrázek 10: Grafické zobrazení příchozích (zelené) a odchozích (červené) paketů při prohlížení internetu




Změřená přenosová rychlost opět odpovídá prohlížení internetu, jsou vidět prudké výkyvy, což je normální, když už je překlad doménového jména dokončen a stránka načtena, obrázek 11.

Obrázek 11: Grafické zobrazení přenosové rychlosti v reálném čase při prohlížení internetu




### 5.3.2 Sledování videa na internetu

Jako další test bylo vybráno měření síťových parametrů při sledování videa uloženého na serveru. Nejprve opět zachycené pakety Snortem s IP adresami, které opět obsahují validní VŠB adresy a komunikace probíhala přes zabezpečený protokol HTTPS - port 443, obrázek 12.

Obrázek 12: Zachycené pakety při sledování videa na internetu

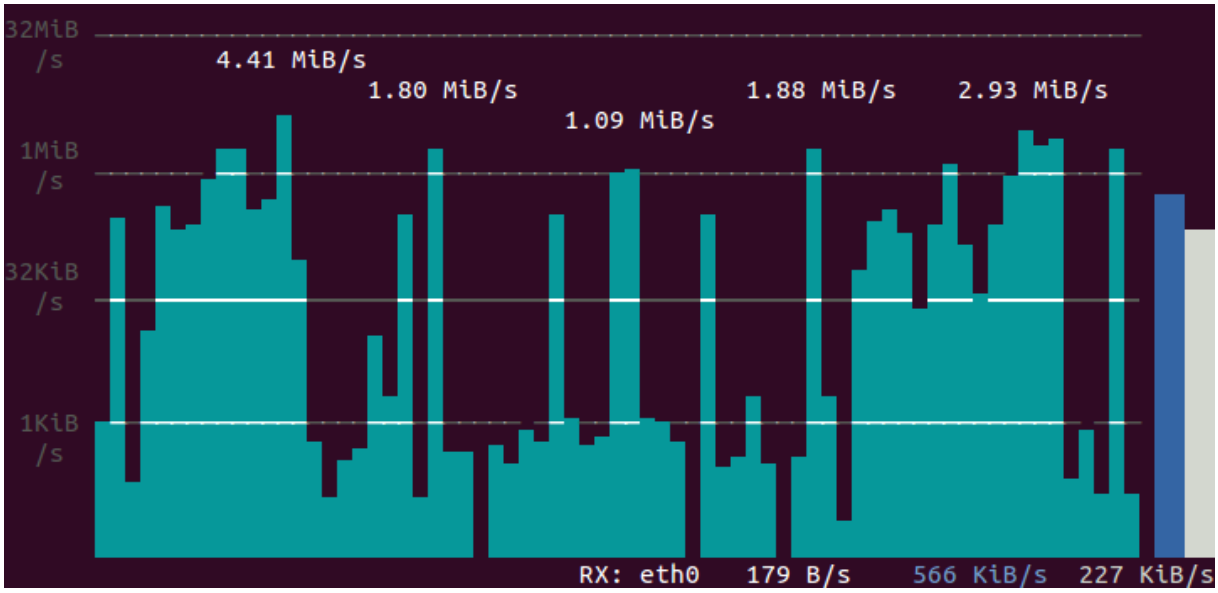
```
04/05-13:16:43.540160 2001:718:1001:2c8:e4cc:566a:5b6b:7573:59431 -> 2001:718:1:
23::c:443
TCP TTL:64 TOS:0x0 ID:0 IpLen:40 DgmLen:72
***A**** Seq: 0xDDE2561D  Ack: 0x46FDDF87  Win: 0x15BB  TcpLen: 32
TCP Options (3) => NOP NOP TS: 224711 1958238148
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/05-13:16:43.540280 2001:718:1:23::c:443 -> 2001:718:1001:2c8:e4cc:566a:5b6b:7
573:59431
TCP TTL:60 TOS:0x0 ID:0 IpLen:40 DgmLen:381
***AP*** Seq: 0x46FDDF87  Ack: 0xDDE2561D  Win: 0x594  TcpLen: 32
TCP Options (3) => NOP NOP TS: 1958238148 224708
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Dále grafické zobrazení odchozích a příchozích paketů Slurmem, obrázek 13.

Obrázek 13: Grafické zobrazení odchozích (červené) a příchozích (zelené) paketů při sledování videa na internetu


Opět je vidět, že nepřišlo žádné neobvyklé množství paketů najednou, ani spoustu paketů pouze našim směrem. Co se přenosové rychlosti v reálném čase týče, ta je rapidně vyšší, než při prohlížení internetu, což je pochopitelné, prudké výkyvy jsou způsobeny překlikáváním časového ukazatele na videu, kdy je třeba data stáhnout a načíst rychleji do paměti, aby mohlo být video přehráváno. Grafické zobrazení na obrázku 14.

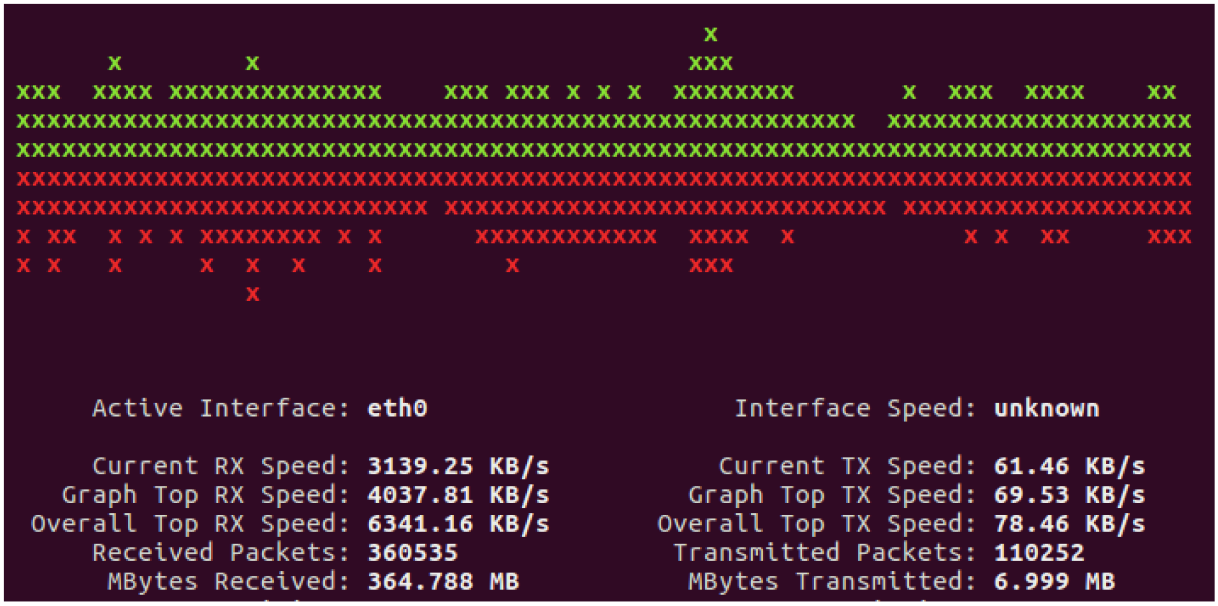
Obrázek 14: Grafické zobrazení přenosové rychlosti při sledování videa na internetu



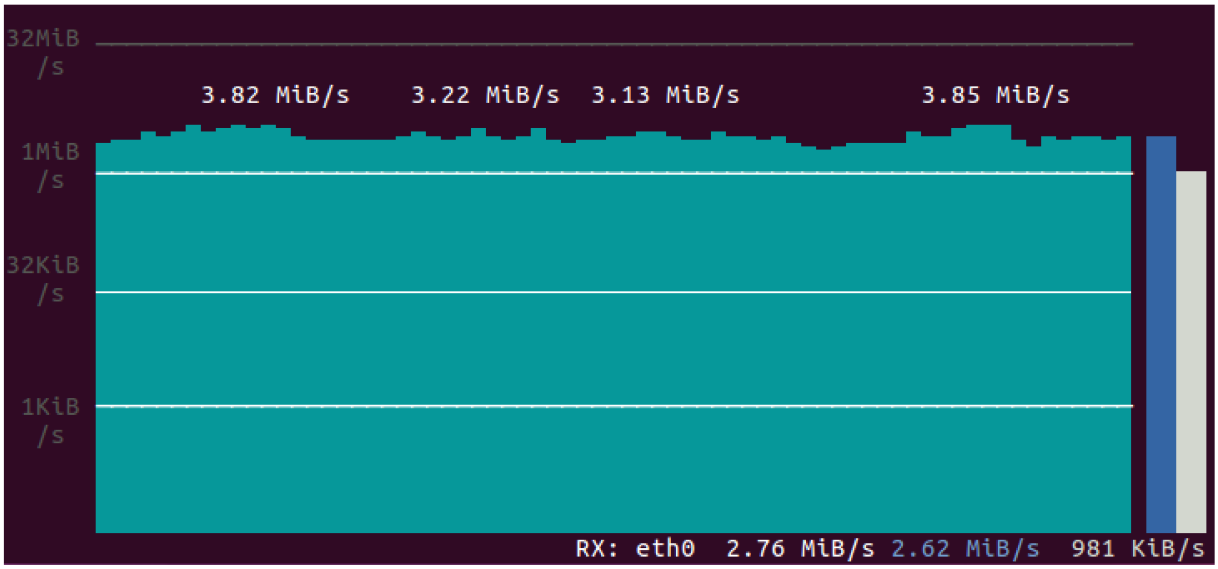

### 5.3.3 Stahování souboru

Tento test je pravděpodobně nejdůležitější, neboť se jeho chování nejvíce podobá spoustě útoků na síť, pomineme-li IP adresy, které při útoku pravděpodobně nebudou validní (můžou být, pokud je útočník velice zručný a dokázal zduplikovat nám známou IP adresu nebo pokud útok pochází z vnitřní sítě). Odchozí a příchozí pakety na obrázku 15 a přenosová rychlost 16. Je tedy vidět téměř konstantní přenosová rychlost a počet vyslaných a přijatých paketů je na grafu v danou chvíli(ve statistikách ne, protože jsou zaznamenány všechny pakety, které byly zachyceny i během ostatních testů) také téměř totožný (objem vyslaných a přijatých dat však nikoliv).

Obrázek 15: Grafické zobrazení odchozích (červené) a příchozích (zelené) paketů při stahování souboru




Obrázek 16: Grafické zobrazení přenosové rychlosti při stahování souboru




## 5.4 Útok hrubou silou

Útoky hrubou silou patří podle statistik k těm nejčastějším. V tomto případě byl simulován útok, který měl za cíl zjistit heslo k připojení na stanici pomocí SSH. Nejprve je potřeba zjistit, jestli je port 22, tedy SSH otevřený, pomocí nmapu. Pokud je, je možno zahájit pokus o zjištění hesla.

K tomuto účelu byl použit program Hydra. Z počítače s IP adresou 2001:db8:ab:2::3 byl tedy zahájen pokus o zjištění hesla (na tomto počítači byl soubor se spoustou hesel, které program postupně zkoušel) k připojení na počítač 2001:db8:ab:2::1. Byl použit příkaz:

```
hydra -s 22 -l root -P '/var/pwds.txt' -v -t 128 2001:db8:ab:2::1 ssh
```

Útok sice úspěšný nebyl, nepodařilo se získat heslo, nicméně Snort rozpoznal síťové anomálie, pravidlo bylo napsáno následovně:

```
alert tcp any any -> 2001:db8:ab:2::/64 22 (msg:"SSH utok hrubou silou";
flow:to_server,established; content:"SSH-"; depth:4; detection_filter:track
    by_src, count 30, seconds 60; sid:14500;)
```

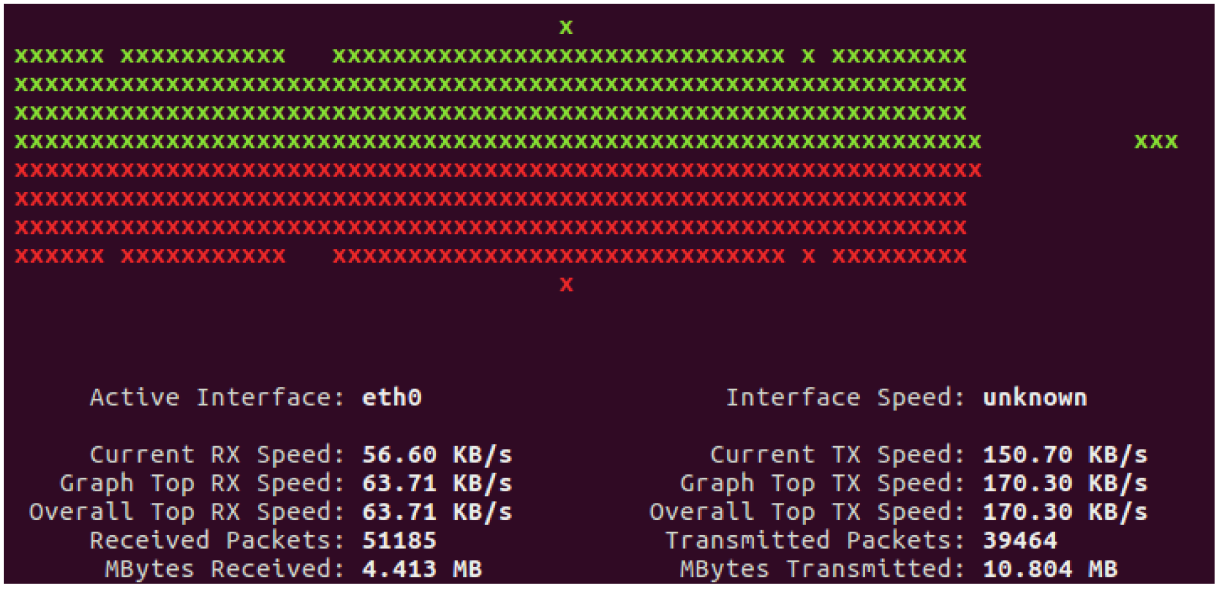
Pravidlo tedy kontroluje připojení na port 22 - SSH, navázané TCP spojení a obsah paketu "SSH", tato situace musí nastat 30x během 60 sekund, aby byla vyhodnocena jako anomálie. Výstup ze Snortu na obrázku 17.

Obrázek 17: Zachycený paket Snortem

```
[**] [1:14500:0] SSH utok hrubou silou. [**]
[Priority: 0]
04/08-11:07:32.258998 2001:db8:ab:2::3:44078 -> 2001:db8:ab:2::1:22
TCP TTL:64 TOS:0x0 ID:0 IpLen:40 DgmLen:94
***AP*** Seq: 0xD488494C  Ack: 0xA9933759  Win: 0x384  TcpLen: 32
TCP Options (3) => NOP NOP TS: 965099 975443
```

Útok Hydra postupně inkrementuje číslo portu a zkouší se připojit pokaždé z jiného. Pokus o zjištění byl úspěšně zaznamenán i programem Slurm, obrázek 18

Obrázek 18: Odeslané (červené) a přijaté (zelené) pakety při útoku hrubou silou




Také Speedometer zaznamenal příliš jednotvárnou rychlost podobající se spíše stahování souboru při konstantní rychlosti, obrázek 19.

Obrázek 19: Přenosová rychlost při útoku hrubou silou


## 5.5 Nestandardní obsah paketu

Tento test byl proveden za účelem zjištění funkcionality vyhledávání dat v paketech. Do paketu bylo vloženo slovo "virus". Pochopitelně ve skutečnosti útočník pravděpodobně takto neprozradí pochybná data, nicméně některé viry mají známé názvy souborů, a tak toto testování jako příklad posloužilo dobře. Pravidlo ve Snortu jsem napsal následovně:

```
alert IP any any -> 2001:db8:ab:2::/64 any (msg:"Mozny prenos viru na 64bit
    system Ubuntu"; content:"Ubuntux64virus"; depth:100; sid:15666;)
alert IP any any -> 2001:db8:ab:2::/64 any (msg:"Mozny prenos viru"; content:"
    virus"; depth:100; sid:15888;)
```

Pravidlo je tedy napsáno tak, aby Snort prohledal datovou část paketu do hloubky 100 znaků. V tomto případě nebylo třeba monitorovat přenosovou rychlost, či sledovat vzorec zasílání paketů, bylo nutné prohledat data v paketech. Pakety byly odeslány z počítače s adresou 2001:db8:ab:2::3 na počítač s adresou 2001:db8:ab:2::1 a byly vytvořeny v programu Packet sender, poté odeslány, obsah je vidět ve Sniffer režimu, obrázek 20.

Obrázek 20: Pochybný paket s obsahem "virus"

```
04/10-21:04:23.501412 08:00:27:9F:4D:D5 -> 08:00:27:28:BF:A4 type:0x86DD len:0xAC
2001:db8:ab:2::3:56834 -> 2001:db8:ab:2::1:14456 UDP TTL:64 TOS:0x0 ID:0 IpLen:40 DgmLen:158
Len: 110
2E 2E 2E 2E 2E 2E 2E 2E 2E 30 30 30 30 30 30  ..........000000
30 35 35 35 35 35 35 35 35 6B 6B 6B 6B 6B 6A 64  055555555kkkkkjd
6A 73 6A 64 6A 73 64 6A 73 6B 6B 6B 34 35 38 37  jsjdjsdjskkk4587
35 34 36 38 37 36 35 34 36 35 34 55 62 75 6E 74  54687654654Ubunt
75 78 36 34 76 69 72 75 73 61 64 6B 61 73 64 6B  ux64virusadkasdk
61 73 6B 64 61 6B 73 64 6B 61 73 6B 64 37 38 34  askdaksdkaskd784
35 36 34 39 38 36 35 34 34 36 31 33 32 34        56498654461324
```

Snort podle napsaných pravidel v NIDS režimu úspěšně zachytil pochybný obsah, obrázek 21.

Obrázek 21: Snortem vygenerované hlášení při odchycení pochybných paketů

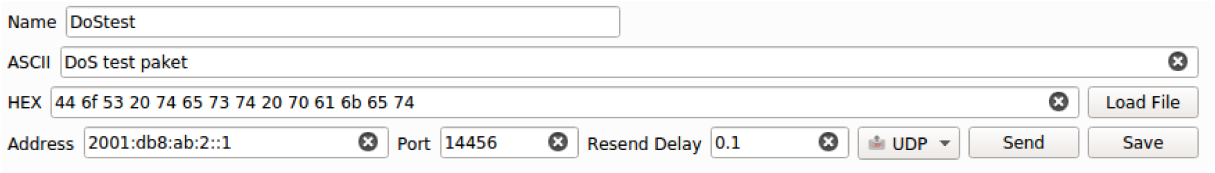
```
[**] [1:15888:0] Mozny prenos viru [**]
[Priority: 0]
04/10-21:10:58.348369 08:00:27:9F:4D:D5 -> 08:00:27:28:BF:A4 type:0x86DD len:0xAC
2001:db8:ab:2::3:56834 -> 2001:db8:ab:2::1:14456 UDP TTL:64 TOS:0x0 ID:0 IpLen:40 DgmLen:158
Len: 110

[**] [1:15666:0] Mozny prenos viru na 64bit system Ubuntu [**]
[Priority: 0]
04/10-21:10:59.124550 08:00:27:9F:4D:D5 -> 08:00:27:28:BF:A4 type:0x86DD len:0xAC
2001:db8:ab:2::3:56834 -> 2001:db8:ab:2::1:14456 UDP TTL:64 TOS:0x0 ID:0 IpLen:40 DgmLen:158
Len: 110
```

## 5.6 DoS útok

Jako další útok způsobující síťové anomálie byl zvolen druhý nejčastější útok, tedy DoS. Tomuto typu útoku je věnována největší pozornost, protože je poměrně jednoduché jej použít a pokud je v systému špatně nakonfigurovaný firewall nebo není přítomen NIDS program, nemusíme jej odhalit, pochopitelně je tento útok později pocítit tím nejhorším možným způsobem a to

shozením běžících služeb či procesů a následným zamrznutím systému. Nejprve bylo programem Packet sender posílány pakety v intervalu 10 paketů/s 22.

Obrázek 22: Nastavení UDP paketu v programu Packet sender




V tomto testu nešlo o to zasílat velké množství paketů najednou, ale pouze o zasílání paketů při konstantní rychlosti bez navázání spojení, abych otestoval Snort. Pravidlo bylo napsáno následovně:

```
alert udp any any -> 2001:db8:ab:2::/64 any (msg:"Mozny UDP DoS utok."; flow:
    stateless; sid:12455;)
```

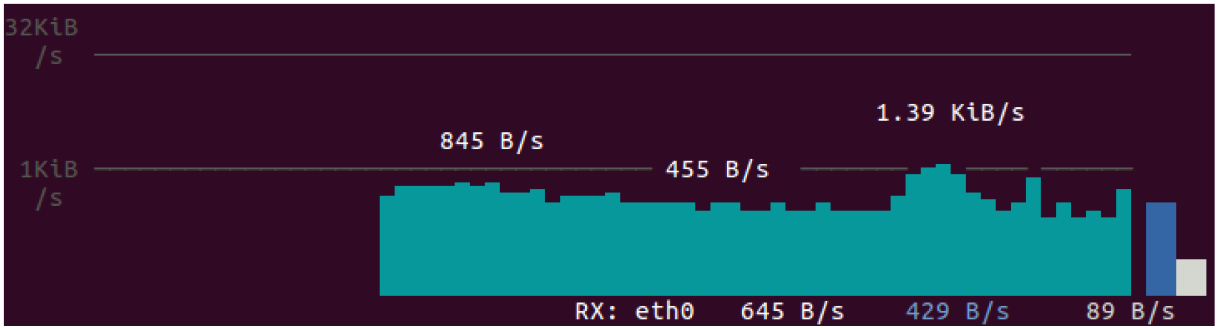
Tento test pochopitelně nijak nenarušil běh OS ani sítě samotné, nicméně potenciální útok byl rozeznatelný podle vzorců v grafech a Snort zachytil UDP pakety zasílané podobným způsobem, jako u DoS útoků, obrázek 23.

Obrázek 23: Snortem zachycený UDP paket použitý pro test vyhledávání řetězců

```
[**] [1:12455:0] Mozny UDP DoS utok. [**]
[Priority: 0]
04/09-12:36:02.138908 2001:db8:ab:2::2:53253 -> 2001:db8:ab:2::1:14456
UDP TTL:64 TOS:0x0 ID:0 IpLen:40 DgmLen:62
Len: 14
```

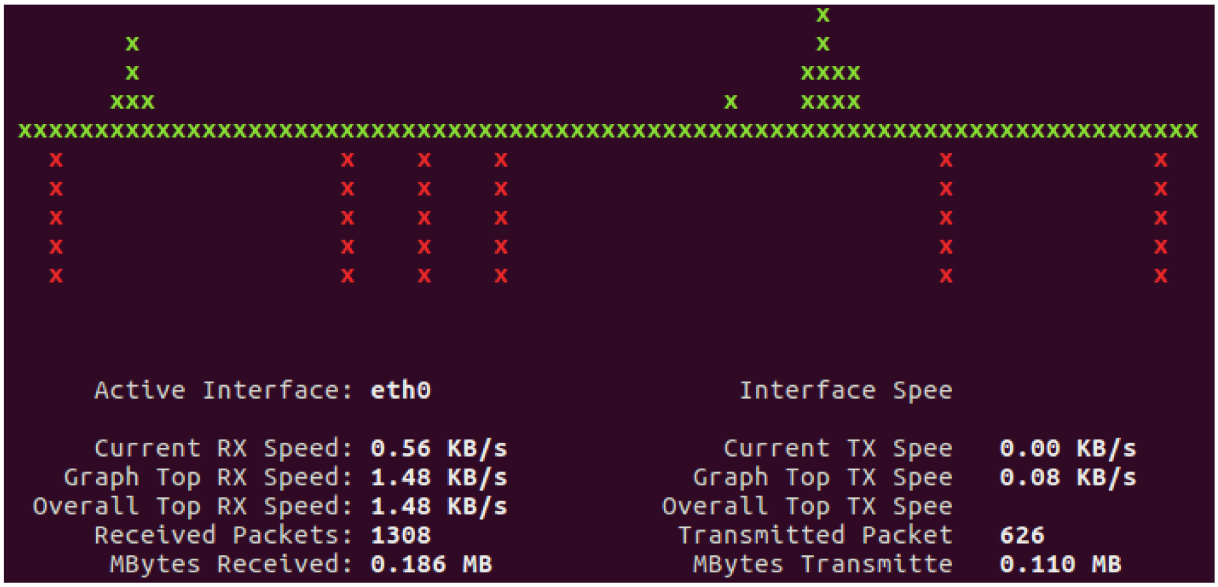
Co se přenosové rychlosti týče, také byla zaznamenána anomálie, tento vzorec by odpovídal stahování dat při omezené rychlosti (běžná praktika, využíváno například u herních klientů, kdy se stahují nová data na pozadí při malé rychlosti, aby nebyl narušen chod hry při síťové hře). Opět nic stahováno nebylo a tak lze usoudit, že se může jednat o nějaký útok, obrázek 24.

Obrázek 24: Grafické zobrazení přenosové rychlosti při odesílání 10 UDP paketů za sekundu




Ve Slurmu lze vidět opět velice malý objem přenášených dat, nicméně více směřují pakety naším směrem, obrázek 25, což odpovídá vygenerovaným paketům.

Obrázek 25: Grafické zobrazení odchozích (červené) a příchozích (zelené) paketů při odesílání 10 UDP paketů za sekundu




Nyní už k reálnému DoS útoku. Proběhl z počítače 2001:db8:ab:2::3 na počítač 2001:db8:ab:2::1. Na to byl použit router flood útok, speciálně pro IPv6 síť, který zneužívá RA, zasílá tedy ICMPv6 pakety přes linkové adresy v obrovském množství s informacemi, které říkají, že je potřeba nastavit IPv6 adresy (při stahování souboru bylo zachyceno cca 3500 paketů za sekundu, při tomto útoku přibližně 6500 paketů za sekundu). Pro útok byla použita distribuce Linuxu Backtrack 5 R3 a příkaz je:

```
flood_router6 eth0
```

Napsané pravidlo ve Snortu:

```
alert icmp any any -> 2001:db8:ab:2::/64 any (msg:"ICMP DoS utok"; threshold:
    type both, track by_dst, count 300, seconds 60; sid:102544;)
```

Pravidlo je napsáno tak, aby tyto ICMP pakety zachytával jako anomálii jen v případě, nastane-li tato situace více jak 300x za 60 sekund. Klasický ping nastane 60x za 60 sekund, proto je zde hranice posunuta výš. Paket zachycený Snortem 26.

Obrázek 26: Paket zachycený Snortem při router flood DoS útoku

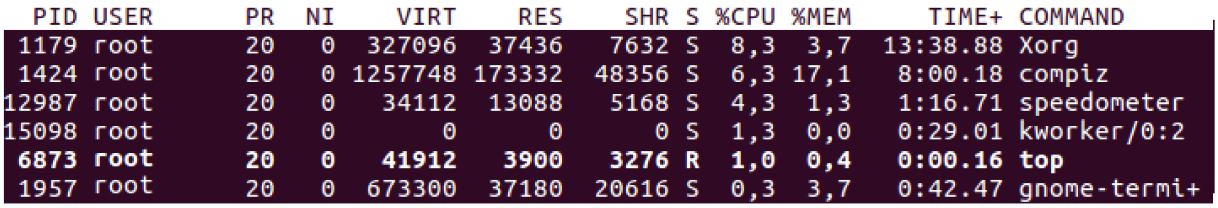
```
[**] [1:102544:0] ICMP DoS utok. [**]
[Priority: 0]
04/08-11:41:24.048108 fe80::218:82ff:fee5:a977 -> ff02::1
IPV6-ICMP TTL:255 TOS:0x0 ID:0 IpLen:40 DgmLen:104
```

Je zřetelně vidět ICMP paket a používání linkových adres, přes které se realizuje komunikace při RA a RS, čehož tento útok zneužívá. Průběh zasílání a přijímání paketů při flood útoku vypadá téměř totožně, jako při útoku na SSH pomocí programu Hydra (lišila se jen přenosová rychlost 27)18.

Obrázek 27: Grafické zobrazení přenosové rychlosti při flood útoku


Nicméně žádná z těchto statistik není sama o sobě pro systém či síť nijak kritická, pokud v dnešní době není někde internet o rychlosti pod 1Mb/s. Tento útok se dá odhalit, jenže až se tak stane, většinou už splnil, co měl, v tomto případě zaměstnal procesor, a to kriticky. Zatížení procesoru při běžném provozu 28.

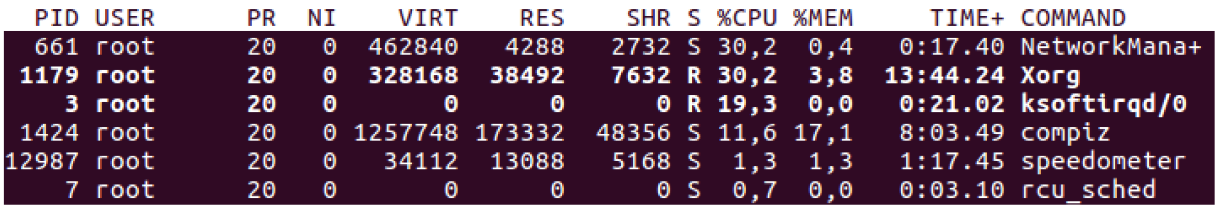
Obrázek 28: Vytížení procesoru před DoS útokem



| PID | USER | PR | NI | VIRT | RES | SHR | S | %CPU | %MEM | TIME+ | COMMAND |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1179 | root | 20 | 0 | 327096 | 37436 | 7632 | S | 8,3 | 3,7 | 13:38.88 | Xorg |
| 1424 | root | 20 | 0 | 1257748 | 173332 | 48356 | S | 6,3 | 17,1 | 8:00.18 | compiz |
| 12987 | root | 20 | 0 | 34112 | 13088 | 5168 | S | 4,3 | 1,3 | 1:16.71 | speedometer |
| 15098 | root | 20 | 0 | 0 | 0 | 0 | S | 1,3 | 0,0 | 0:29.01 | kworker/0:2 |
| 6873 | root | 20 | 0 | 41912 | 3900 | 3276 | R | 1,0 | 0,4 | 0:00.16 | top |
| 1957 | root | 20 | 0 | 673300 | 37180 | 20616 | S | 0,3 | 3,7 | 0:42.47 | gnome-termi+ |

A zatížení procesoru po úspěšném útoku 29.

Obrázek 29: Vytížení procesoru po DoS útoku



| PID | USER | PR | NI | VIRT | RES | SHR | S | %CPU | %MEM | TIME+ | COMMAND |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 661 | root | 20 | 0 | 462840 | 4288 | 2732 | S | 30,2 | 0,4 | 0:17.40 | NetworkMana+ |
| 1179 | root | 20 | 0 | 328168 | 38492 | 7632 | R | 30,2 | 3,8 | 13:44.24 | Xorg |
| 3 | root | 20 | 0 | 0 | 0 | 0 | R | 19,3 | 0,0 | 0:21.02 | ksoftirqd/0 |
| 1424 | root | 20 | 0 | 1257748 | 173332 | 48356 | S | 11,6 | 17,1 | 8:03.49 | compiz |
| 12987 | root | 20 | 0 | 34112 | 13088 | 5168 | S | 1,3 | 1,3 | 1:17.45 | speedometer |
| 7 | root | 20 | 0 | 0 | 0 | 0 | S | 0,7 | 0,0 | 0:03.10 | rcu_sched |

Bylo zjištěno, že po útoku je procesor vytížen téměř na maximum a systém nezvládá plynule zpracovávat ani základní úkony, jako spouštění textového souboru či prohlížení internetu. Je vidět vysoká zátěž u Network Managera, který nestíhá zpracovávat požadavky routeru, dále Xorg, což je v podstatě základ pro grafické rozhraní, které spravuje více monitorů, či jiné externí zařízení, jako je klávesnice či myš v systému Ubuntu, nicméně se mi nepodařilo zjistit, proč tento útok vytížil právě proces Xorg. Poslední více vytížený proces je ksoftirqd, což je kernelové vlákno, které se nachází na každém jádře procesoru a spustí se tehdy, když je na systém spoustu lehce systém narušujících požadavků, což tento útok splňuje. Pro zajímavost, program neběžel ani 1000ms a počítač, na který byl útok cílený si už na rozhraní eth0 vytvořil spoustu adres, obrázek 30.

Obrázek 30: Rozhraní eth0 po DoS útoku

```
eth0      Link encap:Ethernet  HWaddr 08:00:27:28:bf:a4
          inet6 addr: 2a01:2bea:2fcf:38be:a22b:dd5f:35e2:2a4f/64 Scope:Global
          inet6 addr: 2001:db8:ab:2::1/64 Scope:Global
          inet6 addr: 2a01:599a:3b5d:af03:5101:7cb0:a4ea:98cb/64 Scope:Global
          inet6 addr: 2a01:e677:70fc:b66a:1e76:31a1:f718:f4b/64 Scope:Global
          inet6 addr: fe80::bc41:3a05:8718:3ac2/64 Scope:Link
          inet6 addr: 2a01:947f:cf:dcb0:57dc:c440:e5c5:3e97/64 Scope:Global
          inet6 addr: 2a01:2e96:c2:fc6d:9742:bbe4:158c:5c82/64 Scope:Global
          inet6 addr: 2a01:e677:70fc:b66a:5101:7cb0:a4ea:98cb/64 Scope:Global
          inet6 addr: 2a01:75e9:9876:b874:5101:7cb0:a4ea:98cb/64 Scope:Global
          inet6 addr: 2a01:42a4:c930:c005:5101:7cb0:a4ea:98cb/64 Scope:Global
          inet6 addr: 2a01:16a1:9466:6c00:5101:7cb0:a4ea:98cb/64 Scope:Global
          inet6 addr: 2a01:ee42:ed1c:3cfd:5622:5a34:e1ba:e809/64 Scope:Global
          inet6 addr: 2a01:8e16:153e:fa91:5101:7cb0:a4ea:98cb/64 Scope:Global
          inet6 addr: 2a01:8e16:153e:fa91:f53d:8bc2:6432:58ca/64 Scope:Global
          inet6 addr: 2a01:e6f9:315:c83c:5101:7cb0:a4ea:98cb/64 Scope:Global
          inet6 addr: 2a01:947f:cf:dcb0:5101:7cb0:a4ea:98cb/64 Scope:Global
          inet6 addr: 2a01:3b87:250e:5c03:5101:7cb0:a4ea:98cb/64 Scope:Global
          inet6 addr: 2a01:2bea:2fcf:38be:5101:7cb0:a4ea:98cb/64 Scope:Global
          inet6 addr: 2a01:e6f9:315:c83c:a6b:afa3:83bb:e0ed/64 Scope:Global
          inet6 addr: 2a01:a5e2:1d6e:12de:5101:7cb0:a4ea:98cb/64 Scope:Global
          inet6 addr: 2a01:16a1:9466:6c00:cb25:e3a5:55a1:92f8/64 Scope:Global
          inet6 addr: 2a01:b76f:d3d5:dee5:832c:6003:4037:24b6/64 Scope:Global
          inet6 addr: 2a01:a5e2:1d6e:12de:4a66:73f6:f867:ea01/64 Scope:Global
          inet6 addr: 2a01:599a:3b5d:af03:b01e:d4e:392d:7c3b/64 Scope:Global
          inet6 addr: 2a01:56ef:e37b:fd3f:c928:50f:92b9:7926/64 Scope:Global
          inet6 addr: 2a01:42a4:c930:c005:3560:bb8d:12c4:afa2/64 Scope:Global
          inet6 addr: 2a01:3b87:250e:5c03:6dc7:fefd:f877:26fb/64 Scope:Global
          inet6 addr: 2a01:e3b9:b0e3:7bac:fbd1:72ee:8183:7a71/64 Scope:Global
          inet6 addr: 2a01:75e9:9876:b874:ecba:a6d:2bde:f881/64 Scope:Global
          inet6 addr: 2a01:b76f:d3d5:dee5:5101:7cb0:a4ea:98cb/64 Scope:Global
          inet6 addr: 2a01:56ef:e37b:fd3f:5101:7cb0:a4ea:98cb/64 Scope:Global
          inet6 addr: 2a01:2e96:c2:fc6d:5101:7cb0:a4ea:98cb/64 Scope:Global
          inet6 addr: 2a01:e3b9:b0e3:7bac:5101:7cb0:a4ea:98cb/64 Scope:Global
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:22712 errors:0 dropped:0 overruns:0 frame:0
          TX packets:151 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2683139 (2.6 MB)  TX bytes:20749 (20.7 KB)
```

# 6 Závěr

Cílém této práce bylo zjistit, co to vlastně jsou síťové anomálie, způsoby vyhledávání a monitorování těchto anomálií. Nejprve byly popsány sítě jako takové, poté už jednotlivé anomálie, respektive útoky, které je způsobují. Dále byla provedena analýza rozdílů a změn u IPv6 sítě, protože měření probíhalo ve školní laboratoři, která funguje čistě na protokolu IPv6. Dále byly rozepsány jednotlivé způsoby jak anomálie vyhledávat. Praktickými cíli této práce bylo vybrat a nakonfigurovat linuxové programy určené jak pro detekci anomálií, tak pro jejich vytvoření. Program Snort byl popsán nejdetailněji, včetně konfigurace. Dále byly popsány programy, které vizualizovaly síťový provoz a přenosovou rychlost. V neposlední řadě byly zmíněny programy na generování paketů a programy, které byly využity k vytváření útoků na síť. Posledním a nejdůležitějším bodem této práce bylo zátěžové testování v laboratorních podmínkách. Byla provedena řada testů, od bezpečnostních průniků, jako jsou neoprávněné připojení přes SSH, či skenování portů, přes útok hrubou silou až po DoS útok. Útoky byly popsány, jak byly započaty, co způsobily a jakým způsobem je zachytit Snortem. Dále bylo pro porovnání provedeno testování síťového provozu při běžných činnostech, jako je prohlížení internetu a stahování souboru, aby bylo možno rozlišit provoz při útoku a bez něj. Ke všem útokům jsou doloženy obrázky paketů a síťového provozu. Byla tedy zjištěna bezproblémová funkčnost Snortu v IPv6 síti, ovšem s útoky to bylo horší, neboť většina známých útoků byla cílena pouze na IPv4 sítě, nicméně všechny útoky uvedeny v této práci jsou v IPv6 síti realizovatelné. Díky této práci jsem zjistil, jak reálné útoky odhalovat a jak je generovat. Téma, které mi bylo vybráno panem vedoucím mě velmi zaujalo a v budoucnu, například v diplomové práci, či v budoucím zaměstnání, bych se tímto odvětvím, tedy bezpečností v počítačových sítích, chtěl zabývat více a hlouběji.

Jakub Večerík

# Literatura


[1] Snort users manual 2.9.9 Dostupný z: https://www.snort.org/documents/snort-users-manual

[2] BHATTACHARYYA, Dhruba K. Network anomaly detection: a machine learning perspective. ISBN 978-1-4665-8208-8.

[3] Monowar H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, Network Anomaly Detection: Methods, Systems and Tools

[4] BEALE, Jay, Andrew R. BAKER a Joel. ESLER. Snort: IDS and IPS toolkit. Burlington, MA: Syngress, c2007. ISBN 978-1-59749-099-3.

[5] How to test Snort. [online]. Dostupné z: http://www.computerweekly.com/tip/How-to-test-Snort

[6] Gaia Maselli, Luca Deri, Stefano Suin, Design and Implementation of an Anomaly Detection System: an Empirical Approach Dostupný z: http://luca.ntop.org/ADS.pdf

[7] nmap(1) - Linux man page. Linux Documentation [online]. Dostupné z: https://linux.die.net/man/1/nmap

[8] Bezpečnostní mechanismy – IPv6.cz. [online]. Dostupné z: https://www.ipv6.cz/Bezpe%C4%8Dnostn%C3%AD_mechanismy

[9] nmap(1) - Linux man page. Linux Documentation [online]. Dostupné z: https://linux.die.net/man/1/nmap

[10] denial of service - How a DOS TCP packet different from normal Packet? - Information Security Stack Exchange. Information Security Stack Exchange [online]. Dostupné z: https://security.stackexchange.com/questions/34003/how-a-dos-tcp-packet-different-from-normal-packet

[11] IDS. Systémy detekce průniku v Linuxu [online]. Dostupné z: http://www.cs.vsb.cz/grygarek/SPS/projekty0405/IDS/ids.html

[12] How to test Snort. ComputerWeekly.com | Information Technology (IT) News, UK IT Jobs, Industry News [online]. Dostupné z: http://www.computerweekly.com/tip/How-to-test-Snort

[13] BackTrack Linux - Penetration Testing Distribution [online]. Copyright © BackTrack Linux 2017 [cit. 13.04.2017]. Dostupné z: http://www.backtrack-linux.org/

[14] V čem se IPv6 liší a na co bychom si při jeho implementaci měli dát pozor - CSIRT. CSIRT.CZ - CSIRT [online]. Dostupné z: https://www.csirt.cz/page/3099/v-cem-se-ipv6-lisi-a-na-co-bychom-si-pri-jeho-implementaci-meli-dat-pozor/

[15] What are good IDS testing tools? : sysadmin. reddit: the front page of the internet [online]. Copyright © 2017 reddit inc. All rights reserved. [cit. 17.04.2017]. Dostupné z: https://www.reddit.com/r/sysadmin/comments/xi13l/what_are_good_ids_testing_tools/

[16] Detecting Network Anomalies using Traffic Modeling [online]. Copyright © [cit. 17.04.2017]. Dostupné z: http://www.cs.colostate.edu/~cs557/Slides/19AnomalyDetection.pdf

[17] Sample Default Rules | Working with Snort Rules | InformIT. InformIT: The Trusted Technology Source for IT Pros and Developers [online]. Copyright © 2017 Pearson Education, [cit. 17.04.2017]. Dostupné z: http://www.informit.com/articles/article.aspx?p=101171&seqNum=11

[18] networking - How to display network traffic in terminal - Ask Ubuntu. Ask Ubuntu [online]. Dostupné z: https://askubuntu.com/questions/257263/how-to-display-network-traffic-in-terminal